




LAB-2

Terraform AWS Provider and IAM User Setting

Step 1: Create a new directory

```
~ as   
→ mkdir terraform  
  
~ as   
→ cd terraform  
  
~/terraform as   
→
```


Step 2: IAM user

User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☒ Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

 **Are you providing console access to a person?**

User type

☐ Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

☒ I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Set permissions


Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☒ **Add user to group**
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ **Copy permissions**
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ **Attach policies directly**
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

 **Get started with groups**
Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

Create group

► **Set permissions boundary - optional**

Cancel Previous Next

Review and create


Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name lab-ex	Console password type Custom password	Require password reset Yes
---------------------	--	-------------------------------

Permissions summary

< 1 >

Name 	Type	Used as
IAMUserChangePassword	AWS managed	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

✔ User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

View user



[IAM](#) > [Users](#) > Create user

Step 1

[Specify user details](#)

Step 2

[Set permissions](#)

Step 3

[Review and create](#)

Step 4


Retrieve password


Retrieve password


You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Email sign-in instructions 

Console sign-in URL
 <https://339712729715.signin.aws.amazon.com/console>

User name
 lab-ex

Console password
 ***** [Show](#)

Cancel

Download .csv file

Return to users list

Step 3: Create Terraform configuration file

```
main.tf
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.35.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "ap-south-1"
12   access_key = "
13   secret_key = "
14 }
```

Step 4: Initialize Terraform

```
X terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.35.0"...
- Installing hashicorp/aws v5.35.0...
- Installed hashicorp/aws v5.35.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

~/terraform v1.7.2 default as took 10s
→
```